# LLM-gesteuerte Agenten in Computerspielen

Woche Eins

Von **Cedric Beck** und **Felix Koppe**

# Agenda

- Einleitung
- Spielidee
- Moodboard
- Architektur
- Herausforderungen
- Technik
- Ausblick

# Einleitung

- Zuversichtlich, dass **Echtzeitinteraktion** möglich ist
- Aber: Latenz bleibt kritisch
- Hardware als begrenzender Faktor

# Spielidee: Kernaspekte

- Die Möglichkeiten von LLMs ausschöpfen
  - Dynamische Reaktionen auf das Verhalten des Spielers
  - Breit gefächerte Geschichte mit **minimalen Einschränkungen**
  - Sprache muss zentrale Rolle spielen
- Den Spieler durch Entscheidungsdruck und Zeitlimitationen fordern
  - Fokus auf Immersion und eine intensive Erfahrung
  - Der Spieler spricht frei

# Spielidee: "Standoff"

- Der Spieler spielt den Kriesenmanager während einer Geiselnahme
  - Er verhandelt mit den Geiselnehmern per **Funk** oder **Telefon**
  - Maßnahmen führen zur Konfliktlösung oder Eskalation
- Der Spieler muss sich entscheiden was er wann macht
- Dem Spieler stehen Live-Überwachungsbilder zur Verfügung
  - (händisch vorgefertigt)

# Spielidee: Potenzielle Features

- Der Spieler muss eine **Pressekonferenz** halten
  - Reporter stellen kritische Fragen zum Krisenmanagement
  - Nachrichtenkanal überträgt die Konferenz in Echtzeit
  - Das Verhältniss zu den Geiselnahmen wird durch die Aussagen des Spielers beeinflusst
- **Polizeidatenbank** durchsuchen (gefüllt mit hilfreichen und irrelevanten Datensätzen)
- Telefonate (mit Informanten oder Polizeipräsident)
- Kollegen benötigen Anweisungen

# Spielidee: Schauplatz

- Fiktive Stadt (erzählerische Freiheiten)
- Freistehendes Gebaude (**Bank** oder **Supermarkt**)
- Von Polizei Abgesperrt, Schaulustige und **Reporter**
- **Mobile Einsatztzentrale** als "Büro" des Spielers

# Moodboard: Zentrale

# Architektur

- Layer Ansatz
  - **Mehrere Modelle**
  - Ein großes Hauptmodell
  - Mehrere kleinere spezialisierte Modelle
- Neustarts kleinerer Modelle um Halluzinationen zu vermeiden

# Architektur

- **GAMEMASTER** LLM:
  - Größeres Modell
  - Hält die Geschichte erzählerisch beisammen
  - Schreibt anweisungen für die untergeordneten Modelle
- **SLAVE** LLMs:
  - Kleineres modell
  - Verkörpern Personen in Dialogen (Reporter/Geiselnehmer/usw.)
  - hält sich an anweisungen vom GAMEMASTER
  - gibt gezielt Informationen preis oder oder hält sie zurück

# Herausforderungen

- Echtzeitverarbeitung und Reaktionsgeschwindigkeit
- Mehrere modelle
  - Koordination
  - Ressourcen
- Kontrolle der Geschichte
  - Initiale Prompts müssen präzise sein
  - Darstellung muss zu Geschichte passen
- Models wollen nicht "böse" sein

# Technik: Allgemein

- Sollte Open Source sein
- Alles Server spezifische in einen docker container

# Technik: Backends

- llama.cpp
- Ollama (nutzt llamacpp)
- vLLM

# Technik: Text2Speech

- **Text2Speech**
- piper (schnell aber schlechte Qualität)
- OpenVoiceV2 (langsamer und ressourcen intensiv)
- **Coqui** (schnell auf cpu)

# Technik: Speech2Text

- **Speech2Text**
- Whisper (openai)
- Fast-Whispe (x4 speed)

# Technik: Modelle

- Flexibel austauschbar
- Wichtige Aspekte
  - Geschwindigkeit
  - Rollenspiel Fähigkeiten
- Mögliche Modelle:
  - mixtral
  - deepseek-r1

# Latenz

Abschätzung

# Ausblick

Fragen

Arbeitsplan